Math 4710/6710 – Graph Theory – Fall 2019

Assignment 3

Usual instructions:

- Solutions to problems should be fully explained, using clear English sentences where necessary.
- Solutions should be written (or typed) neatly on one side only of clean paper with straight (not ragged) edges.
- Multiple pages should be stapled (not clipped or folded) together.
- Problems are from the newer printings of the textbook. If the problem has a different number in older printings, this will be indicated by '(old x.y.z)'.

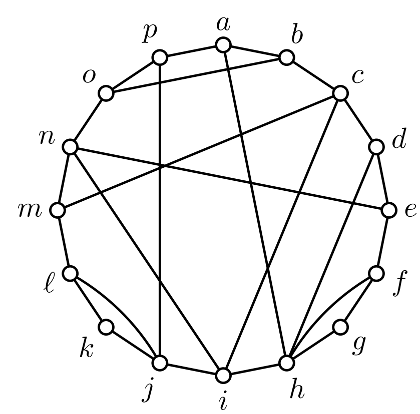
Special instructions:

- Part A is regular problems. They may be typed or handwritten. They are due in class on Friday, October 11, and will be graded once by the instructor. All students should do 3A1, 3A2 and 3A3. Math 6710 students should also do 3A4.
- The website contains a file that has copies of the graphs for problems 3A1, 3A2 and 3A3 so that you can print them out instead of drawing them if you like.
- Part B consists of problems that will be peer-edited. Details are still being worked out! However, all students will be required to submit a first attempt at problems 3B1 and 3B2 by 1 p.m. on Wednesday, October 9. Solutions must be typed. (You may use any suitable software to do this. I will provide some information about using LaTeX if you wish to use that.) Solutions will be submitted electronically. (I have not yet decided whether this will be done by email or using Brightspace.)

Part A

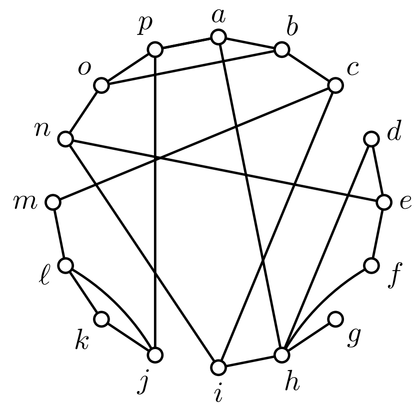
3A1. Using the algorithm described in class, find a BFS tree rooted at a in the graph shown at right below.

Process the neighbours of each vertex in alphabetical order. List the vertices in the order they were added to the tree, along with the time at which each was added. Mark the edges of your tree on a copy of the graph, and label each vertex with its level in the tree. Draw a nice copy of your final tree separate from the graph, with the vertices at each level on the same horizontal line, occurring left to right according to the time when they were found.

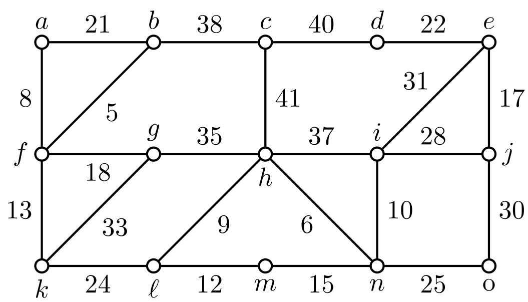


3A2. Using the algorithm described in class, find a DFS tree rooted at a in the graph shown at right below. (Note: this is NOT the same as the graph in 3A1 above.)

Process the neighbours of each vertex in alphabetical order. Show the initial (arrival) and final (departure) times for each vertex. Mark the edges of your tree on a copy of the graph, and label each vertex with its level in the tree. Draw a nice copy of your final tree separate from the graph, with the vertices at each level on the same horizontal line, occurring left to right according to the time when they were found. (Since DFS trees tend to be long and thin, you may alternatively draw the tree sideways by reflecting it in the line $y = -x$, with the vertices at each level on the same vertical line, occurring top to bottom according to the time when they were found.)



3A3. For the graph shown below, find a minimum weight spanning tree by using (i) Kruskal's algorithm, and (ii) the Jarník-Prim algorithm, starting at h. In each case show the edges of the tree on the graph, and provide a list of the edges of the tree in the order in which you added them to the tree.



3A4, Math 6710 students only, modified version of B&M (2nd pr.) 3.4.5. (a) Use Rédei's Theorem (Section 2.2 of B&M) to show that every tournament with three or more vertices is either strong or can be transformed into a strong tournament by the reversal of just one arc.

(b) Is this true for tournaments with one or two vertices?

Part B

3B1, B&M (2nd pr.) 3.2.4. Let G be a k-regular bipartite graph with $k \geq 2$. Show that G has no cutedge.

3B2, adapted from B&M (2nd pr.) 2.1.11(b). A *topological sort* of a digraph D is a linear ordering of its vertices such that, for every arc a of D, the tail of a precedes its head in the ordering. Show that a digraph admits a topological sort if and only if it is acyclic. (You may find it helpful to use the fact that an acyclic digraph always has a source and a sink.)